Task Specific Kinodynamic Motion Planning using Differential Flatness for Aerial Manipulation

Skylar X. Wei[1], Peder Hårderup[2], Thomas Maxfield[3], and Joel W. Burdick[1]

***Abstract*— The dynamics of a general class of aerial manipulators, equipped with an arbitrary k-linked manipulator, are differentially flat. While we provide a new proof of flatness, this paper primarily focuses on flatness-inspired kinodynamic motion planning for aerial manipulators. Using the flat outputs, we introduce a method to plan unit-time kinematically feasible paths between equilibrium state end-effector poses, while satisfying state constraints. Using time-dilation on this path, we determine a trajectory whose velocities, accelerations, and torques satisfy given constraints. An offline nonlinear programming formulation, which uses the above trajectory as an initial guess, optimizes mission objectives, such as minimum kinetic energy and traverse time. The proposed motion planner is demonstrated experimentally in two configurations: (1) a 4 degree of freedom (DoF) manipulator equipped with an end-effector camera for reconnaissance, and (2) a 3 DoF manipulator with a gripper for precisely grasping and arranging objects.**

I. Introduction

An aerial manipulator consists of an underactuated multi-rotor with a multi-link manipulator attached mainly, but not always, at the body's geometric center. Aerial manipulators are of increasing interest [1] since they inherit the high mobility of conventional multi-rotors, with the added ability to interact with the environment via the robot arm's end-effector. They have many practical applications, such as delivering packages [2], inspection of physical infrastructure using arm-mounted sensors [3][4], and tool operation [5]. Despite their potential, aerial manipulators present many challenges in trajectory planning and control. Poorly planned maneuvers and manipulations can be kinodynamically infeasible or unstable due to their underactuated nature and dynamic coupling effects from moving heavy arms or payloads.

This paper provides a new proof that a general class of aerial manipulators are *differentially flat*. Flat systems are equivalent to a trivial system via an endogenous transformation [6], which enables dynamic feedback linearization. Hence, our flatness results certify that this class of system is locally nonlinear controllable, and more importantly enables new kinodynamic trajectory planners, which are the main focus of this paper. Common trajectory planning methods are based on the fact that multi-rotors (without robot arms) are differentially flat [7], [8]. Initial efforts to prove the differential flatness of aerial manipulators required limiting

[1]Engineering and Applied Sciences, California Institute of Technology, Pasadena, CA, USA. {swei, jburdick}@caltech.edu.

[2]Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden. {pederhar}@kth.se.

[3]Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, CA, USA. {tmaxfield}@ucla.edu

The work is partially funded by NASA Jet Propulsion Laboratory and California Institute of Technology Graduate Student Fellowship.

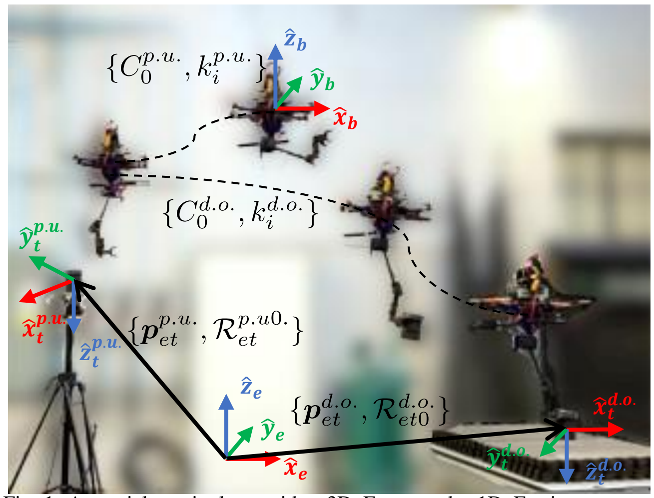
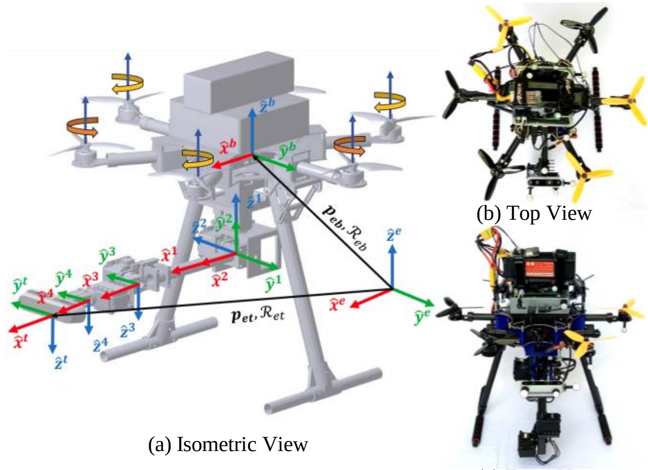


Fig. 1: An aerial manipulator with a 3DoF arm and a 1DoF gripper grasps and arranges objects. The object's pick up (p.u.) location and graspable orientation $\{\boldsymbol{p}_{et}^{p.u.}, \mathcal{R}_{et}^{p.u.}\}$ and its drop off (d.o.) location and orientation $\{\boldsymbol{p}_{et}^{d.o.}, \mathcal{R}_{et}^{d.o.}\}$ are known by the aerial manipulator. The aerial manipulator plans a kinodynamically feasible polynomial trajectory characterized by $\{C_0^{p.u.}, k_i^{p.u.}\}$ and $\{C_0^{d.o.}, k_i^{d.o.}\}$ using the proposed flat output and pre-solve algorithm to achieve object pick up and drop off.

assumptions. E.g., [9] showed that an aerial manipulator with a 2-DoF arm is differentially flat, but assumed that the center of mass (CoM) must be fixed in the end-effector frame, which implies a motionless arm. A planar aerial manipulator with rigid or elastic joints was proven to be flat in [10], and this result was generalized to protocentric manipulators in [11]. However, the overall system CoM must be fixed, else there are unaccounted Coriolis terms. Welde $et.al.$ [12] recently showed that a large class of aerial manipulators posses the flatness property. We show that the same flat output can be easily obtained from the D'Alembert principle.

Kinodynamic motion planning [13] incorporates differential constraints [14], such as velocity and acceleration bounds. For differentially flat systems, control input bounds can be expressed in terms of flat outputs and their derivatives, thereby converting them to kinodynamic constraints. Our main contribution is explicit algorithms to generate kinodynamic trajectories that satisfy given tasks while respecting state, velocity, and control constraints (which were not considered in [12]). Control-aware planners for aerial manipulators [15], [16] rely on sample-based techniques that may include these constraints, yet lack experimental validation.

Leveraging flatness, we propose an optimization-based planning algorithm that produces kinodynamically feasible trajectories that connect one hovering configuration to another, while satisfying tool frame constraints at given waypoints. Using the complete system dynamics, the flat output trajec-



(a) Isometric View

(b) Top View

(c) Front View

Fig. 2: Hardware and annotated view of the aerial manipulator system equipping a 4 DoF manipulator and a Intel RealSense D435i as its tool.

tory is mapped into conventional states for hardware implementation. The full nonlinear optimal problem is infeasible for online computation. Instead, we pre-solve a feasible, but sub-optimal trajectory for online use, which can also serve as an initial condition for off-line nonlinear optimization. We demonstrate the method on hardware performing autonomous reconnaissance and precision grasping tasks.

II. System Description

We analyze the class of aerial manipulators seen in Fig. 2.

Definition: An **Aerial Manipulator (AM)** consists of:

- A multirotor with n ($n \geq 2$) pairs of identical rotors attached to a common base. Each pair contain a clockwise and a counterclockwise rotating rotor. All thrust axes point in a common direction, denoted by $\hat{z}_b$.
- A k-link serial chain manipulator linkage attached to the base's geometric center with the tool being its k^{th} joint. The joints are revolute or prismatic.
- All system components are rigid, and complex-fluid structure interactions are ignored.

Our model is derived using the following reference frames:

- The earth-fixed inertial frame $E = \{O^e, \hat{\boldsymbol{x}}_e, \hat{\boldsymbol{y}}_e, \hat{\boldsymbol{z}}_e\}$.
- The aerial-base body frame $B = \{O^b, \hat{\boldsymbol{x}}_b, \hat{\boldsymbol{y}}_b, \hat{\boldsymbol{z}}_b\}$.
- Manipulator i^{th} link frame $L_i = \{O^i, \hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i, \hat{\boldsymbol{z}}_i\}$.
- End Effector or Tool frame $T = \{O^t, \hat{\boldsymbol{x}}_t, \hat{\boldsymbol{y}}_t, \hat{\boldsymbol{z}}_t\}$.
- Instantaneous CoM frame $C = \{O^c, \hat{\boldsymbol{x}}_c, \hat{\boldsymbol{y}}_c, \hat{\boldsymbol{z}}_c\}$.

Notationally, $\boldsymbol{p}_{ab} \in \mathbb{R}^3$ and $\mathcal{R}_{ab} \in SO(3)$ denote the position and orientation of frame B relative to frame A. We assign robot arm link frames as in Fig. 2. Let $\mathcal{R}_{i-1,i}$ denote the relative rotation of the i^{th} link frame with respect to the $(i-1)^{th}$ link frame, with where $R_{01} = R_{b1} = I_{3\times 3}$. Let $\boldsymbol{\eta} \triangleq [\eta_1, \cdots, \eta_k]^T \in \mathbb{R}^k$ denote the vector of link joint angles. The spatial linear velocity of the aerial-base is denoted as $\dot{\boldsymbol{p}}_{eb}$, and the B frame body angular velocities is ω_b. Using an Euler roll, pitch, and yaw angles $\boldsymbol{\xi} = [\phi, \theta, \psi]^T$ parameterization of $SO(3)$, $\dot{\mathcal{R}}_{eb} \triangleq \mathcal{R}_{eb}\hat{\boldsymbol{\omega}}_b$ where $\hat{(\cdot)}$ is the 3×3 skew-symmetric operator such that $\forall \boldsymbol{\beta} \in \mathbb{R}^3, \hat{\boldsymbol{\omega}}_b\boldsymbol{\beta} = \boldsymbol{\omega}_b \times \boldsymbol{\beta}$. We denote the map between the body frame angular velocity ω_b and $\dot{\boldsymbol{\xi}}$ as Q where $\omega_b = Q(\boldsymbol{\xi})\dot{\boldsymbol{\xi}}$.

The position and orientation of the i^{th} manipulator link frame (relative to the E frame) is denoted as $\boldsymbol{p}_{ei} = \boldsymbol{p}_{eb} + \mathcal{R}_{eb}\boldsymbol{p}_{bi}$ and $\mathcal{R}_{ei} = \mathcal{R}_{eb}\mathcal{R}_{bi}$, respectively. The displacement, $\boldsymbol{p}_{bi}$, and rotation, $\mathcal{R}_{bi}$, of a link i's reference frame relative to the B frame can be computed (in homogeneous coordinates) as:

$$\begin{bmatrix} R_{bi} & p_{bi} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \triangleq H_{0i} = H_{01}H_{12}\cdots H_{(i-1)i}$$

For each arm link, its linear velocity (in B frame) $\dot{\boldsymbol{p}}_{bi}$ and body velocity (in L_i) frame $\dot{\boldsymbol{\omega}}_i$, can be computed as [17],

$$\dot{\boldsymbol{p}}_{bi} = -\dot{\mathcal{R}}_{bi}\mathcal{R}_b{i}^T\boldsymbol{p}_{bi} + \dot{\boldsymbol{p}}_{bi} \quad \text{and} \quad \boldsymbol{\omega}_i = (\mathcal{R}_{bi}^T\dot{\mathcal{R}}_{bi})^\vee,$$

respectively. The $\vee$ isomorphism operator converts skew-symmetric matrices $\hat{\boldsymbol{v}}$ to the equivalent vector $\boldsymbol{v} \in \mathbb{R}^3$.

The system's CoM location is a function of the variables, $\boldsymbol{p}_{ec} = \boldsymbol{p}_{eb} + \mathcal{R}_{eb}\sum_{i=1}^{k}\frac{m_i}{m_t}\boldsymbol{p}_{bi}$, where m_t is the total system mass. Frame C is chosen parallel to frame B: $\mathcal{R}_{bc} = I_{3\times 3}$. The kinetic and potential energy of each rigid body is

$$K_a = \frac{1}{2}m_a\dot{\boldsymbol{p}}_{ea}^T\dot{\boldsymbol{p}}_{ea} + \frac{1}{2}\boldsymbol{\omega}_a^T I_a\boldsymbol{\omega}_a, \quad V_a = m_a g\boldsymbol{e}_3^T\boldsymbol{p}_{ea},$$

The Lagrangian is $\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{a=0}^{k} K_a(\boldsymbol{q}, \dot{\boldsymbol{q}}) - V_a(\boldsymbol{q})$, where $\boldsymbol{q} = [\boldsymbol{p}_{eb}^T, \boldsymbol{\xi}^T, \boldsymbol{\eta}^T]^T$ is a generalized coordinate where $a = 0$ is the rotorbody. We use the Lagrange-D'Alembert principle to obtain the equations of motion (EoM) for the AM,

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\boldsymbol{q}}} - \frac{\partial L}{\partial\boldsymbol{q}} = \mathcal{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathcal{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathcal{G}(\boldsymbol{q}) = \boldsymbol{f}_{gen}, \quad (1)$$

where $\boldsymbol{f}_{gen}$ is the generalized forces, $\mathcal{M}(\boldsymbol{q}), \mathcal{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \mathcal{G}(\boldsymbol{q})$ are the mass matrix, Coriolis matrix, and gravity term, respectively. Moreover, the positive definite mass matrix is only a function of $\boldsymbol{\eta}$, i.e. $\mathcal{M}(\boldsymbol{q}) = \mathcal{M}(\boldsymbol{\eta})$. The generalized forces $\boldsymbol{f}_{gen} = [\boldsymbol{f}_e^T, \boldsymbol{\tau}_{gen}^T, \boldsymbol{\tau}_{\boldsymbol{\eta}}^T]^T$ where $\boldsymbol{f}_e = \mathcal{R}_{eb}\boldsymbol{e}_3 f_t \in \mathbb{R}^3$ and $\boldsymbol{\tau}_{gen} \in \mathbb{R}^3$ are the net forces and torques acting on the system. We denote $f_t \in \mathbb{R}$ as the total thrust and $\boldsymbol{\tau}_{\boldsymbol{\eta}} \in \mathbb{R}^k$ as the input torques of the manipulator joints. The system inputs are $\boldsymbol{u} = [f_t, \boldsymbol{\tau}_b^T, \tau_{\boldsymbol{\eta}}^T]^T$ where $\boldsymbol{\tau}_b$ are the net torques generated by the rotor in B frame which can be mapped it into generalized net torques as $\boldsymbol{\tau}_{gen} = Q(\boldsymbol{\xi})\boldsymbol{\tau}_b$.

Equilibrium Condition: Because the aerial base is underactuated, we focus on trajectories that connect two hoverable, equilibrium states, which allows for a duration-independent analysis. Similar to [18], letting $\dot{\boldsymbol{q}}_{eq} = \ddot{\boldsymbol{q}}_{eq} = 0$ results in zero roll and pitch angles at equilibrium: $\phi_{eq} = \theta_{eq} = 0$, with a total thrust $f_t = m_t g$.

III. Differential Flatness

Flatness [6] concerns systems as a differential field generated by a set of states and inputs. A *flat system* has well understood nonlinear structures which can be exploited for control algorithm design, trajectory generation, and stabilization [19]. We adopt the definition from [20] for characterizing differential flatness for mechanical control systems. Suppose the system with state $\boldsymbol{x}$ and input $\boldsymbol{u}$ has the form,

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \qquad \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{u} \in \mathbb{R}^m, \quad (2)$$

where the map $f(\cdot) \in \mathcal{C}^\infty$ within the given domain. The vector $\boldsymbol{y} \in \mathbb{R}^m$ is a *flat output* if the followings are true,

- $y_i = g_i(\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{u}}, \dots, \boldsymbol{u}^{(j_i)}),\ j_i \in \mathbb{N}, i = 1, 2, \dots, m.$
- $x_i = h_i(\boldsymbol{y}, \dot{\boldsymbol{y}}, \dots, \boldsymbol{y}^{(k_i)}),\ k_i \in \mathbb{N}, i = 1, 2, \dots, n,$
  $u_i = \tilde{h}_i(\boldsymbol{y}, \dot{\boldsymbol{y}}, \dots, \boldsymbol{y}^{(l_i)}),\ l_i \in \mathbb{N}, i = 1, 2, \dots, m.$
- All components of $\boldsymbol{y}$ are differentially independent, i.e. no $\boldsymbol{y}$ that satisfies $\Phi(\boldsymbol{y}, \dot{\boldsymbol{y}}, \dots, \boldsymbol{y}^{(k)}) = 0,\ k \in \mathbb{N}$.

Same as the flat outputs proposed by [12], we provide an alternative proof using standard EoM (1) which contains useful relations for the latter planning section.

Theorem 1. *The flat outputs ($\boldsymbol{\sigma} \in \mathbb{R}^{4+k}$) for a general k-link aerial manipulator consists of the overall CoM position in E frame $\boldsymbol{p}_{ec}$, heading ψ, and manipulator joint angles $\boldsymbol{\eta}$.*

Proof. The only external forces that act on the floating base are the summed rotor thrusts in the $\hat{\boldsymbol{z}}_b$ direction, and the gravitational force in direction $\hat{\boldsymbol{z}}_e$. Thus the CoM linear acceleration must thus satisfy:

$$m_t \ddot{\boldsymbol{p}}_{ec} = -m_t g \boldsymbol{e}_3 + \mathcal{R}_{eb} \boldsymbol{e}_3 f_t. \tag{3}$$

From (3), the thrust magnitude can be found as $f_t = m_t \|\ddot{\boldsymbol{p}}_{ec} + g\boldsymbol{e}_3\|_2$. From the x and y component, accelerations can be arranged to express Euler angles ϕ and θ as a function of the flat output $\boldsymbol{\sigma}$ and its 2^{nd} order time derivatives,

$$\phi = \sin^{-1}\left(\frac{(\ddot{x}_{ec}\sin\psi - \ddot{y}_{ec}\cos\psi)}{\|\ddot{\boldsymbol{p}}_{ec} + g\boldsymbol{e}_3\|_2}\right), \tag{4}$$

$$\theta = \tan^{-1}\left(\frac{\ddot{x}_{ec}\cos\psi + \ddot{y}_{ec}\sin\psi}{\ddot{z}_{ec} + g}\right). \tag{5}$$

Hence, one can show the base frame position is also a function the flat output and its higher derivatives $\boldsymbol{p}_{eb}(\boldsymbol{\sigma}, \ddot{\boldsymbol{\sigma}}) = \boldsymbol{p}_{ec} - \mathcal{R}_{eb}(\boldsymbol{\xi})\frac{1}{m_t}\sum_{i=1}^{k} m_i \boldsymbol{p}_{bi}(\boldsymbol{\eta})$. Using the system state $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{\sigma}, \ddot{\boldsymbol{\sigma}})$ and (1), the remaining system inputs can be expressed using $\boldsymbol{\sigma}$ up to its 4^{th} time derivatives as

$$\begin{bmatrix} \boldsymbol{\tau}_b \\ \boldsymbol{\tau}_{\boldsymbol{\eta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & Q^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & I \end{bmatrix} \left(\mathcal{M}(\boldsymbol{\eta})\ddot{\boldsymbol{q}} + \mathcal{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathcal{G}(\boldsymbol{q})\right). \tag{6}$$

To conclude, the proposed $\boldsymbol{\sigma}$ is a flat output. □

Singularities occur when the system is in free fall or at zero thrust, i.e. $\ddot{z}_{ec} + g = 0$, or if either roll and pitch angle takes value $\pm\pi/2$ rad. Intuitively, the system instantaneously loses one degree of control in these cases, and cannot maintain hover. We restrict the attitude angles to avoid singularity, $\phi, \theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, and thrust to be positive in flight, $f_t > 0$.

IV. Task-based Kinodynamic Planning

We introduce a motion planner that takes desired *tool frame* waypoints $(\boldsymbol{p}_{et}^{des}, \boldsymbol{R}_{et}^{des})$ and generates a time-based trajectory $(\boldsymbol{p}_{eb}(t), \psi(t), \boldsymbol{\eta}(t))$ that moves the multi-rotor from its current equilibrium configuration to a new one where its tool frame aligns with the given waypoint. Most importantly, physical inequality constraints like velocity, acceleration, and force/torque bounds must be satisfied while avoiding known stationary obstacles. We also enforce hover at the beginning and end of the trajectory to enable time-independent tool operation. The common constraints are categorized in Table I by their dependency on the number of state differentials. Inspired by the polynomial-time approximation scheme (PTAS) [21] and a minimum snap formulation [22], we propose an optimization based two-step trajectory planner that uses the flat output $\boldsymbol{\sigma}$ and its relations to states $\boldsymbol{q}$ and input $\boldsymbol{u}$.

Common Aerial Manipulator Constraints

Description	Type	Equation
Corridor Path	State	$\boldsymbol{p}_l + \boldsymbol{d}_{am} \leq \boldsymbol{p} \leq \boldsymbol{p}_u - \boldsymbol{d}_{am}$
Joint Angle	State	$\boldsymbol{\eta}^{min} \leq \boldsymbol{\eta} \leq \boldsymbol{\eta}^{max}$
Physical Roll	State	$-\pi/2 < \phi < \pi/2$
Physical Pitch	State	$-\pi/2 < \theta < \pi/2$
Linear Velocity	Rate	$\dot{\boldsymbol{p}}_{eb}^{min} \leq \dot{\boldsymbol{p}}_{eb} \leq \dot{\boldsymbol{p}}_{eb}^{max}$
Angular Rate	Rate	$\boldsymbol{\omega}_b^{min} \leq \boldsymbol{\omega}_b \leq \boldsymbol{\omega}_b^{max}$
Joint Velocity	Rate	$\dot{\boldsymbol{\eta}}^{min} \leq \dot{\boldsymbol{\eta}} \leq \dot{\boldsymbol{\eta}}^{max}$
Rotor Thrust	Input	$0 \leq f_i \leq f_i^{max}, \quad \forall i \in \mathbb{N}_1^{2n}$
Total Thrust	Input	$0 < f_t \leq f_t^{max}$
Body Torques	Input	$-\boldsymbol{\tau}_b^{max} \leq \boldsymbol{\tau}_b \leq \boldsymbol{\tau}_b^{max}$
Joint Torque	Input	$\boldsymbol{\tau}_{\boldsymbol{\eta}}^{min} \leq \boldsymbol{\tau}_{\boldsymbol{\eta}} \leq \boldsymbol{\tau}_{\boldsymbol{\eta}}^{max}$

TABLE I: Summary of common AM constraints. The corridors constraints are determined by a high level planner. The vector $\boldsymbol{d}_{am} \in \mathbb{R}^3$ is the principle axes of the smallest ellipsoid encapsulating the aerial manipulator at all orientations. The constraint type in the table is chosen based on the number of state time derivatives, where 0^{th} order ones are called state constraints, and first and second order derivative state constraints are distinguished as rate and input constraints, respectively.

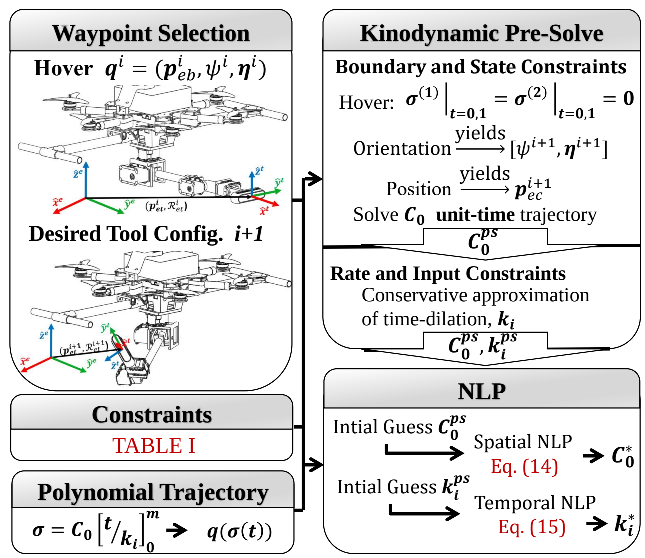


Fig. 3: A pictorial description of our proposed framework. Both the pre-solve and NLP formulation solves a kinodynamically feasible trajectory for a given tool frame way point. The numerically efficient but conservative trajectory from the pre-solve serves as an initial guess for the NLP.

A. Instantaneous Feasible Task Space

Our method applies to aerial manipulators with any number and type of revolute and prismatic joints. In practice, we differentiate among three cases that vary according to the arm's ability to compensate for the underactuated aerial base's inability to roll and pitch at hover. Ignoring joint angle rotation limits, we hereafter assume that the desired tool waypoints are *feasible* according to the definitions below.

Case 1. The aerial manipulator possesses no arm (it uses a fixed boom), or all arm joints are prismatic, or the arm's revolute joint axes align with $\hat{\boldsymbol{z}}_b$. Hence, the tool frame's pitch and roll angle are the same as the aerial base, and the feasible task space is $\{\boldsymbol{p}_{et}, \mathcal{R}(0, 0, \psi)\}$. If the prismatic joint axes $\hat{\boldsymbol{\rho}}_i$ do not align with the aerial-base frame, the feasible task space is $\{\boldsymbol{p}_{et}, \mathcal{R}(0, 0, \psi)\mathcal{R}_{bt}\}$, for some $\mathcal{R}_{bt}$.

Case 2. The arm possesses a single revolute joint with joint axis $\hat{\boldsymbol{\gamma}} \neq \hat{\boldsymbol{z}}_b$. The feasible task space is

$\{\boldsymbol{p}_{et}, \mathcal{R}(0,0,\psi)\mathcal{R}(\hat{\boldsymbol{\gamma}},\eta_1)\}$, where $\mathcal{R}(\hat{\boldsymbol{\gamma}},\eta_1)$ represents rotation about joint axis $\hat{\boldsymbol{\gamma}}$ by angle η_1.

Case 3. The arm contains two or more revolute joints whose axes satisfy $\hat{\boldsymbol{\gamma}}_i \neq \hat{\boldsymbol{z}}_b, \forall i$. Using an angle-axis representation of $SO(3)$, the feasible task space for this configuration is $\{\boldsymbol{p}_{et}, \mathcal{R}_{et}\}$, where $\boldsymbol{p}_{et} \in \mathbb{R}^3, \mathcal{R}_{et} \in SO(3)$.

B. Polynomial Trajectory Representation

Given current configuration $\boldsymbol{q}_i$ and the desired tool frame waypoint $\{\boldsymbol{p}_{et}, \mathcal{R}_{et}\}$, we define a local order $m \geq 4$ polynomial trajectory segments in the flat output $\boldsymbol{\sigma}$,

$$\underbrace{\begin{bmatrix} \sigma_1(t) \\ \sigma_2(t) \\ \vdots \\ \sigma_{4+k}(t) \end{bmatrix}}_{\boldsymbol{\sigma}(t)} = \underbrace{\begin{bmatrix} c^0_{10} & c^0_{11} & \cdots & c^0_{1m} \\ c^0_{20} & c^0_{21} & \cdots & c^0_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c^0_{(4+k)0} & c^0_{(4+k)1} & \cdots & c^0_{(4+k)m} \end{bmatrix}}_{C_0} \begin{bmatrix} 1 \\ t \\ \vdots \\ t^m \end{bmatrix}. \quad (7)$$

Hereafter we denote $[\boldsymbol{t}]_0^m = [1, t, t^2, \cdots, t^m]^T$. For numerical simplicity, we factorize trajectory (7) as $\boldsymbol{\sigma}(C_0, k_i, t) = C_0[\boldsymbol{t_u}/\boldsymbol{k_i}]_0^m$ where $t_u \in [0,1]$ is a unit time polynomial trajectory with a positive, constant time-dilation factor $1/k_i$. The matrix $C_0 \in \mathbb{R}^{4+k \times m}$ dictates the polynomial's geometry. The time-dilation factor “stretches" or “compresses" the trajectory so that its time derivatives can satisfy the bounds of Table I. Further, the k^{th} time derivative of $\boldsymbol{\sigma}$ is,

$$\boldsymbol{\sigma}^{(k)}(C_0, k_i, t) = \prod_{i=0}^{k} C_i \boldsymbol{t}_0^{m-k} = C_0 \prod_{i=1}^{k}(C_i/k_i) \left[\frac{\boldsymbol{t_u}}{\boldsymbol{k_i}}\right]_0^{m-k},$$

where matrices $C_i \in \mathbb{R}^{(m+1)\times(m+1-i)}$ for $i > 1, i \in \mathbb{N}$ are constant. Using (7), we next describe how to represent state, rate, input and current or initial pose (I.V.) and terminal pose (T.V.) constraints via the flat output trajectory.

Initial and Final Value Constraint: Since the time dilation factor only affects the rate of traversal we can enforce the I.V. and T.V. constraints in the unit-time case.

$$\boldsymbol{q}_i = \boldsymbol{q}([c^0_{10}, c^0_{20}, \cdots, c^0_{(4+k)0}]^T), \quad (8)$$

$$\boldsymbol{p}_{et} = \mathcal{R}_{eb}(0,0,\sum_{j=1}^{m} c^0_{4j}) \sum_{i=1}^{k} \frac{m_i}{m_t} \boldsymbol{p}_{bi}(\sum_{i=5}^{4+k}\sum_{j=0}^{m} c^0_{ij}) + \boldsymbol{p}_{eb}(\sum_{i=1}^{3}\sum_{j=0}^{m} c^0_{ij}), \quad (9)$$

$$\mathcal{R}_{et} = \mathcal{R}_{eb}(0,0,\sum_{j=0}^{m} c^0_{4j}) \mathcal{R}_{bt}(\sum_{i=5}^{4+k}\sum_{j=0}^{m} c^0_{ij}). \quad (10)$$

Lemma 1. *Enforcing hover (equilibrium) at time boundaries of a unit time polynomial trajectory of AM is equivalent to enforce* $\boldsymbol{\sigma}^{(1)}|_{t=0,1} = \boldsymbol{\sigma}^{(2)}|_{t=0,1} = \boldsymbol{0}$.

Proof. To achieve $\dot{\boldsymbol{q}}_{eq} = \ddot{\boldsymbol{q}}_{eq} = \boldsymbol{0}$, the first and second derivatives of flat outputs ψ and $\boldsymbol{\eta}$ must be zero evaluated at $t = 0$ and $t = 1$. Further, the equilibrium condition also requires zero pitch and roll $\theta_{eq} = \phi_{eq} = 0$, the resulting base linear velocity and acceleration at $t = 0$ and $t = 1$ becomes,

$$\dot{\boldsymbol{p}}_{eb} = \boldsymbol{0} = \dot{\boldsymbol{p}}_{ec} - \mathcal{R}_{eb}\sum_{i=1}^{k} \frac{m_i \partial \boldsymbol{p}_{bi}}{m_t \partial \boldsymbol{\eta}} \cancelto{\boldsymbol{0}}{\dot{\boldsymbol{\eta}}} - \sum_{i=1}^{k} \mathcal{R}_{eb} \frac{m_i}{m_t} \cancelto{\boldsymbol{0}}{\hat{\boldsymbol{\omega}}_b} \boldsymbol{p}_{bi},$$

and $\ddot{\boldsymbol{p}}_{eb} = \boldsymbol{0} = \ddot{\boldsymbol{p}}_{ec} + \cancelto{\boldsymbol{0}}{(\cdots)}$. Therefore, at $t = 0$ and $t = 1$, $\dot{\boldsymbol{\sigma}} = [\dot{\boldsymbol{p}}_{ec}^T, \dot{\psi}, \dot{\boldsymbol{\eta}}^T] = \boldsymbol{0}$ and $\ddot{\boldsymbol{\sigma}} = [\ddot{\boldsymbol{p}}_{ec}^T, \ddot{\psi}, \ddot{\boldsymbol{\eta}}^T] = \boldsymbol{0}$ $\square$

Using forward kinematics, current position and orientation, we can obtain $\boldsymbol{p}_{ec}(0)$. Further, jointly solving orientation constraint (8) and (10) and Lemma 1 gives $\boldsymbol{\eta}|_{t=1}$ and $\psi|_{t=1}$ which can be used to obtain $\boldsymbol{p}_{ec}(1)$.

State Constraints: To handle a corridor obstacle avoidance constraint, one can first conservatively approximate the safe set as $\tilde{\boldsymbol{p}}_l \leq \boldsymbol{p}_{ec} \leq \tilde{\boldsymbol{p}}_u$ where $\tilde{\boldsymbol{p}}_l = \boldsymbol{p}_l + \boldsymbol{d}_{am} + \boldsymbol{d}_{CoM}$ and $\tilde{\boldsymbol{p}}_u = \boldsymbol{p}_u - \boldsymbol{d}_{am} - \boldsymbol{d}_{CoM}$. Vector $\boldsymbol{d}_{CoM} \in \mathbb{R}^3$ contains the maximum absolute CoM shift in x, y and z across all manipulator configurations. Thus, the shape coefficient of $\boldsymbol{p}_{ec}$ subject to corridor constraint becomes:

$$\tilde{\boldsymbol{p}}_l \leq \begin{bmatrix} \sum_{j=0}^{m} c^0_{1j}[\boldsymbol{i_k \Delta t}]_0^m \\ \sum_{j=0}^{m} c^0_{2j}[\boldsymbol{i_k \Delta t}]_0^m \\ \sum_{j=0}^{m} c^0_{3j}[\boldsymbol{i_k \Delta t}]_0^m \end{bmatrix} \leq \tilde{\boldsymbol{p}}_u, i_k \in \{1, \cdots, \left\lfloor \frac{1}{\Delta t} \right\rfloor\}. \quad (11)$$

where the discrete time step Δt is selected based on the control loop rate. This conservative approximation can be lifted by enforcing obstacle avoidance constraint at every rigid link at the cost of additional $2k\lfloor\frac{1}{\Delta t}\rfloor$ constraints. For the roll and pitch angle constraints, the time-dilation factors k_i are chosen so that the upper and lower bounds for roll and pitch can be enforced at every discrete controlled step,

$$\phi(C_0, \frac{i_k \Delta t}{k_i}), \theta(C_0, \frac{i_k \Delta t}{k_i}) \in (-\frac{\pi}{2}, \frac{\pi}{2}). \quad (12)$$

Since manipulator joint angles are flat outputs, the inequality constraint can be enforced similar to (11).

Input and Rate Constraints: Point-wise enforcement of the total thrust, body torque, and rotor thrust constraints yield a total of $2(2n+4)\lfloor\frac{1}{\Delta t}\rfloor$ inequality constraints, which pose a numerical burden. We offer a tightened reformulation (Lemma 2) which instead only requires $2\lfloor\frac{1}{\Delta t}\rfloor$ constraints.

Lemma 2. *If the flat output trajectory described by* $\{C_0, k_i\}$ *at each discrete time step* $\tau \in \{0, \cdots, \lfloor\frac{1}{\Delta t}\rfloor\}$ *satisfies*

$$\frac{\tilde{f}_t^{min}}{m_t} \leq \|\ddot{\boldsymbol{p}}_{ec}(C_0, k_i, \tau\Delta) + g\boldsymbol{e}_3\|_2 \leq \frac{\tilde{f}_t^{max}}{m_t}, \quad (13)$$

where $\tilde{f}_t^{max} = 2n\min\{f_i^{max}, \tilde{f}_i^{max}\}$ *and* $\tilde{f}_t^{min} = 2n\max\{\tilde{f}_i^{min}\}$*, the theoretical thrusts and torques required to achieve the trajectory* $\{C_0, k_i\}$ *satisfies rotor, total thrust, and body torques constraints described in Table I.*

Proof. The total thrust and body torque constraints can be formulated as convex polytopic constraints on rotor thrusts,

$$\begin{bmatrix} 0 \\ -\tau^{max}_{b,x} \\ -\tau^{max}_{b,y} \\ -\tau^{max}_{b,z} \end{bmatrix} \leq \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ -c_f b & -c_f b & c_f b & c_f b & \cdots \\ c_f b & c_f b & -c_f b & -c_f b & \cdots \\ -c_m b & c_m b & -c_m b & c_m b & \cdots \end{bmatrix}}_{D} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \end{bmatrix} \leq \begin{bmatrix} f_t^{max} \\ \tau^{max}_{b,x} \\ \tau^{max}_{b,y} \\ \tau^{max}_{b,z} \end{bmatrix},$$

where c_f, c_m, and b denote the thrust, rotor moment, and moment arm, respectively. Note that the constant matrix D has full row rank. Using a pseudo inverse, the constraint above can be mapped to individual rotor thrust constraints, denoted as $\tilde{f}_i^{min} \leq f_i \leq \tilde{f}_i^{max}$. Merging these with rotor thrust constraints, we obtain the constraint set

$[\max\{\tilde{f}_i^{min}, 0\}, \min\{\tilde{f}_i^{max}, f_i^{max}\}], \forall i \in \{1, \cdots, 2n\}$. As a tightening reformulation, we define two parallel hyperplanes,

$$S_{min} \triangleq \{f_i | f_1 + f_2 + \cdots + f_{2n} = 2n \max\{\tilde{f}_i^{min}\}\},$$
$$S_{max} \triangleq \{f_i | f_1 + f_2 + \cdots + f_{2n} = 2n \min\{\tilde{f}_i^{max}\}\}.$$

Note, hyperplanes S_1, S_2 and every parallel hyperplane in between have a non-empty intersection with the feasible space of total thrust, body torque, and rotor thrust constraints. Thus, if the polynomial trajectory evaluated at discrete times satisfies $\tilde{f}_t^{min} \leq m_t \|\ddot{\boldsymbol{p}}_{ec}(C_0, k_i, \tau\Delta t) + g\boldsymbol{e}_3\|_2 \leq \tilde{f}_t^{max} \ \forall \tau$, there exist a feasible thrusts for this trajectory. $\square$

Below, we exploit the trajectory's polynomial structure to reduce these constraints down to 2 constraints. Rate constraints (e.g. roll and pitch constraints) are satisfied by point-wise evaluating and adjusting the time-dilation constants (12).

C. Energy Optimal Trajectory Generation via NLP

We now detail a constrained nonlinear optimization program that finds an energy optimal trajectory in two steps. A *spatial optimization* minimizes a unit-time kinetic energy cost under state, I.V., and T.V. constraints. Once a feasible obstacle-free unit-time path has been found, a *temporal optimization* yields a time-optimal time-dilation factor k_i^* that satisfies the remaining constraints. These two steps yield a feasible trajectory that is optimized for the system dynamics.

Spatial NLP

$$C_0^* = \underset{C_0 \in \mathbb{R}^{(4+k)\times m}}{\arg\min} \sum_{i_k=0}^{\lfloor 1/\Delta t \rfloor} \dot{\boldsymbol{\sigma}}^T W \dot{\boldsymbol{\sigma}} \tag{14a}$$
$$\text{s.t.} \quad (8), (9), (10), (11)$$
$$\dot{\boldsymbol{\sigma}}(C_0, 1, 0) = \dot{\boldsymbol{\sigma}}(C_0, 1, 1) = \mathbf{0}, \tag{14b}$$
$$\boldsymbol{\eta}^{min} \leq \begin{bmatrix} \sum_{j=0}^{m} c_{5j}^0 (i_k \Delta t)^j \\ \vdots \end{bmatrix} \leq \boldsymbol{\eta}^{max}, \tag{14c}$$

where $\dot{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}}(C_0, 1, i_k\Delta) = C_0 C_1 [\boldsymbol{i_k \Delta t}]_0^{m-1}$ are the velocity of the flat output states. $W \in \mathbb{R}^{4+k \times 4+k}$ is a user-selected positive definite weight matrix. The cost function (14a) is interpreted as minimizing the kinetic energy rotorcraft CoM and manipulator kinetic energy within unit time.

Temporal NLP

$$k_i^* = \min_{k_i \in \mathbb{R}_{>0}} k_i \tag{15a}$$
$$s.t. \quad (12), (13), \quad \forall \tau \in \{1, \cdots, \lfloor 1/\Delta t \rfloor\},$$
$$\text{Rate Constraints}(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}}, \ddot{\boldsymbol{\sigma}}, \boldsymbol{\sigma}^{(3)}, \boldsymbol{\sigma}^{(4)})|_{t=\frac{\tau\Delta t}{k_i}} \tag{15b}$$
$$\text{Input Constraints}(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}}, \ddot{\boldsymbol{\sigma}}, \boldsymbol{\sigma}^{(3)}, \boldsymbol{\sigma}^{(4)})|_{t=\frac{\tau\Delta t}{k_i}}. \tag{15c}$$

Since initial guesses are crucial for NLPs, we next introduce a pre-solve which quickly generates a feasible initial paths.

D. Kinodynamic Pre-Solve

Complex dynamic coupling can cause numerically intensive trajectory optimization. Numerous efforts have studied sub-optimal trajectories for moving obstacle avoidance [23] and feasible time-optimal avoidance of stationary obstacles [24]. We similarly propose a sub-optimal pre-solve that enforces kinodynamic constraints. We prioritize numerical efficiency over optimality to obtain real-time feasible trajectories.

From the I.V. and T.V. orientation constraints (8) and (10), we note that the tool frame orientation waypoint in hover is only dictated by the heading angle and arm joint angles. Like an inverse kinematics problem, we solve a simple nonlinear program to obtain a feasible trajectory for $\boldsymbol{\eta}$ and ψ,

$$\begin{bmatrix} c_{40}^0 & \cdots & c_{4m}^0 \\ \vdots & \ddots & \vdots \\ c_{(4+k)0}^0 & \cdots & c_{(4+k)m}^0 \end{bmatrix}^{ps} = \underset{c_{ij} \in \mathbb{R}}{\arg\min} \sum_{i=4}^{4+k} \sum_{j=0}^{m} c_{ij}^0 \tag{16}$$
$$\text{s.t.} \quad (8), (10), (14b), (14c).$$

From the manipulator joint angles and the forward kinematics, we obtains shape coefficients for the system CoM,

$$\begin{bmatrix} c_{10}^0 & c_{11}^0 & \cdots & c_{1m}^0 \\ c_{20}^0 & c_{21}^0 & \cdots & c_{2m}^0 \\ c_{30}^0 & c_{31}^0 & \cdots & c_{3m}^0 \end{bmatrix}^{ps} = \underset{c_{ij} \in \mathbb{R}}{\arg\min} \sum_{i=1}^{3} \sum_{j=0}^{m} c_{ij}^0 \tag{17}$$
$$\text{s.t.} \quad (8), (9), (11), (14b).$$

Thus, by combining conditions I.V. and T.V., and the state constraints, we may obtain a feasible unit-time polynomial trajectory described by the shape matrix C_0^{ps}.

The polynomial trajectory exhibits a "Bang–bang" like acceleration where it first gradually accelerates and then decelerates within the unit time. Given the pre-solve feasible shape matrix C_0^{ps}, we can find the minimum and maximum 2^{nd} derivative occurs in $t \in [0, 1]$, denoted as $t_{min}^{(2)}$ and $t_{max}^{(2)}$. The time-dilation factor can be conservatively approximated by solving the nonlinear root-finding problem,

$$\frac{\tilde{f}_t^{min}}{m_t} - g \leq \left\| C_0 \prod_{i=1}^{2} (C_i / k_i) \begin{bmatrix} \boldsymbol{t_{max}^{(2)}} \\ \boldsymbol{k_i} \end{bmatrix}_0^{m-2} \right\|_2 \leq \frac{\tilde{f}_t^{max}}{m_t} - g,$$

which further reduces the original $(2n+4)\lfloor 1/\Delta t \rfloor$ input inequality constraints to just 2 inequality constraints. By the following tightened inequality constraints, the pitch and roll angle constraints are equivalently enforced,

$$\left\| C_0 \prod_{i=1}^{2} (C_i / k_i) \begin{bmatrix} \boldsymbol{t_{max}^{(2)}} \\ \boldsymbol{k_i} \end{bmatrix}_0^{m-2} \right\|_2 \leq g. \tag{18}$$

Joint velocity constraints can be similarly enforced. However, angular velocity constraints are rather difficult to simplify. We recommend either solving a nonlinear optimization problem similar to the temporal NLP (15) or use a heuristic bisection search to find the corresponding time-dilation factor. Lastly, given the time-dilation factors obtained from all constraints, we choose the largest one to ensure feasibility.

V. Hardware Validation

Including the constraints, the pre-solve stage computes a feasible coefficient matrix C_0^{ps} and time-dilation factor k_i^{ps}in in ~0.5 seconds, using uncompiled MATLAB code. We benchmarked this performance with a single-board companion flight computer (DeskMini UM350) powered by an AMD Ryzen 5 3550H processor and 16 GB RAM.

We validated our planner, summarized in Fig.3, on a set of tasks executed by a custom aerial manipulator in two configurations. We perform all flight tests in an indoor arena equipped with an Optitrack motion capture system that

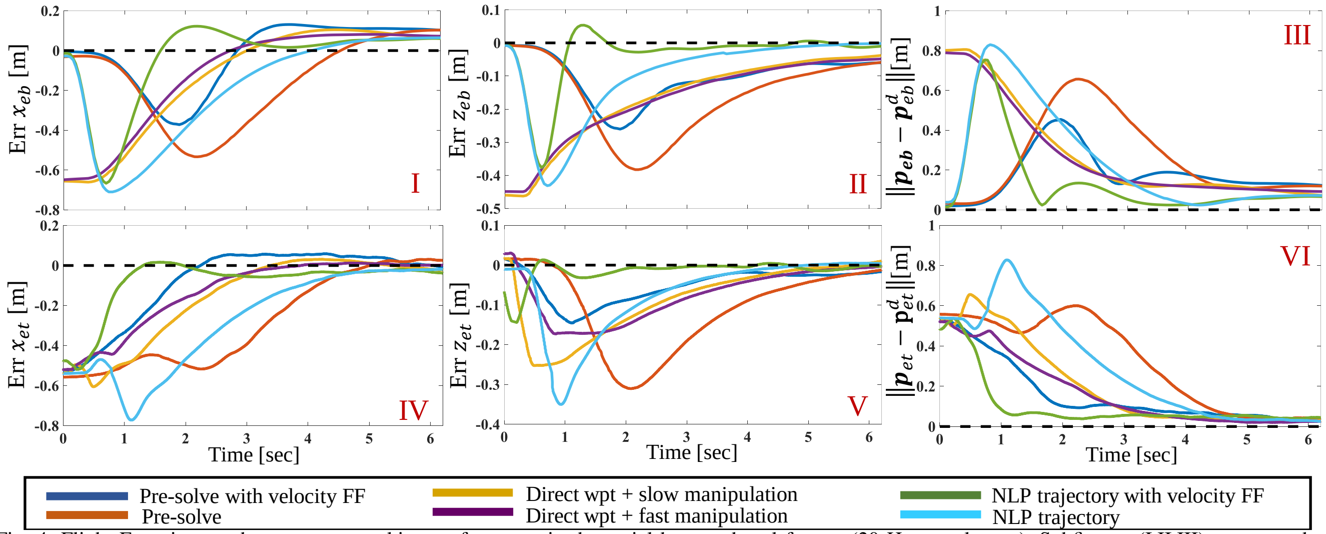


Fig. 4: Flight Experiments that compare tracking performance in the aerial base and tool frames (20 Hz sample rate). Subfigures (I,II,III) corresponds to the tracking error of x-position, z-position, and the 2-norm of aerial base position as it translates between two configurations. Subfigures (IV,V,VI) shows the x and z tracking errors, as well as the 2-norm of tool frame position tracking error under the same waypoint.

samples rotor-craft pose at 190 Hz. These poses are used by the flight autopilot. The main flight controller is a Hex Cube Orange running a PX4 autopilot [25]. The manipulator, attached at the rotor-craft's geometric center, is actuated by DYNAMIXEL MX servos. The onboard/companion computer uses a ROS master node to administer the trajectory setpoints. A safeguard tether attaches to the multirotor base and a $\sim 100\,\text{g}$ passive weight eliminate slack which has minimal interference to flight behavior. The overall system weighs 3.2 kg. Both demonstrations can be viewed with link `https://youtu.be/WB4M359KVIQ`.

Task 1: 4DoF Reconnaissance. A 4DoF manipulator (Fig.2) carries an Intel RealSense D435i camera to demonstrate reconnaissance tasks. Given a sequence of desired tool (camera view) way points, we generated 9^{th} order polynomial paths $\{C_0^{ps}, k_i^{ps}\}_{i=1}^m$ using the kinodynamic pre-solve, as well as $\{C_0^*, k_i^*\}_{i=1}^m$ using the spatial and temporal NLP. In addition to the constraints described in Table I, we also enforced zero jerk and zero snap at the terminal waypoint. We publish the open-loop trajectories at 20 Hz, which are executed by the PX4 controller. Fig. 4 compares a single waypoint tracking performance for the pre-solve and NLP trajectories. The plots also include performance under feedforward (FF). Lastly, the waypoint (wpt) results are obtained from kinematics paths, and do not consider any dynamic couplings. Note that the maximum manipulator joint velocity in the fast case is double that of the slow case. The results in Fig.4 show that the NLP trajectory with velocity FF yields the smallest tracking errors for both tool and aerial base frames. Like [26], we found that feed forward improves tracking accuracy. Also note that the pre-solve time dilation factors are significantly larger than those of the temporal NLP, yielding an equilibrium-like trajectory. On the contrary, the agile NLP trajectory better utilizes the system's thrust and torque capability.

Task 2: 3DoF Gripper. We also performed an object picking and delivering task (Fig. 1). We replaced the camera with a 1DoF gripper. The aerial manipulator autonomously picks up and drops off objects with predetermined feasible waypoints. We observed minor oscillations and drifts in the tool frame as the vehicle approaches the task end point. In future work, we aim to implement a nonlinear model predictive controller (NMPC) that uses the flat trajectory to perform expansive local linearization during the tool operation phase. A specialized controller for body rates thrust control (i.e., replacing the PX4 controller) should improve tracking performance, with faster disturbance rejection.

VI. Conclusion and Future Work

Based upon flatness, we proposed an optimization based task specific trajectory generation method that produces kinodynamically feasible paths that satisfy practically important constraints, and can optimize mission criteria. This method maps the flat trajectories to the standard hardware states of rotorcraft position, heading angle, and manipulator joint angles. We also provide a feasible trajectory *pre-solve* that allows real-time motion planning applications or that can serve as an initial guess to an NLP that jointly minimizes kinetic energy and travel time. Finally, we demonstrated our motion planner as it executed different tasks on flight hardware. In a 4Dof reconnaissance task the kinodynamically feasible trajectory reduces tracking error in comparison to waypoint following. Feedforward of the velocity obtained from the flatness relations improves tracking performance. Moreover, we showed the planner's potential for interactive tasks, such as precision grasping and manipulation. Future work aims to develop flatness-based NMPC and hybrid formulations which are better suited to physical interactive tasks. To avoid non-stationary obstacles, risk-aware planners [27], [28] may be incorporated into the spatial constraints.

VII. Acknowledgement

The authors thank Patrick Spieler and Matthew Anderson for hardware help. This research was funded in part by the Jet Propulsion Laboratory, California Institute of Technology.

References

[1] X. Ding, P. Guo, K. Xu, and Y. Yu, "A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems," *Chinese Journal of Aeronautics*, vol. 32, no. 1, pp. 200–214, 2019.

[2] H. Sayyaadi and A. Soltani, "Modeling and control for cooperative transport of a slung fluid container using quadrotors," *Chinese Journal of Aeronautics*, vol. 31, no. 2, pp. 262–272, 2018.

[3] S. Hamaza, I. Georgilas, G. Heredia, A. Ollero, and T. Richardson, "Design, modeling, and control of an aerial manipulator for placement and retrieval of sensors in the environment," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1224–1245, 2020.

[4] A. Jimenez-Cano, G. Heredia, and A. Ollero, "Aerial manipulator with a compliant arm for bridge inspection," in *2017 Intern. Conf. on Unmanned Aircraft Systems*, pp. 1217–1222, 2017.

[5] H.-N. Nguyen, C. Ha, and D. Lee, "Mechanics, control and internal dynamics of quadrotor tool operation," *Automatica*, vol. 61, pp. 289–301, 2015.

[6] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Differential flatness and defect: An overview," *Banach Center Pub.s*, vol. 32, 01 1995.

[7] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE Int. Conf. Robotics and Automation*, pp. 2520–2525, 2011.

[8] S. Formentin and M. Lovera, "Flatness-based control of a quadrotor helicopter via feedforward linearization," in *IEEE Conf. on Decision and Control and European Control Conf.*, pp. 6171–6176, 2011.

[9] J. Welde and V. Kumar, "Coordinate-free dynamics and differential flatness of a class of 6DOF aerial manipulators," in *2020 IEEE Intern. Conf. on Robotics and Automation*, pp. 4307–4313, 2020.

[10] B. Yüksel, N. Staub, and A. Franchi, "Aerial robots with rigid/elastic-joint arms: Single-joint controllability study and preliminary experiments," in *2016 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems*, pp. 1667–1672, 2016.

[11] B. Yüksel, G. Buondonno, and A. Franchi, "Differential flatness and control of protocentric aerial manipulators with any number of arms and mixed rigid-/elastic-joints," in *2016 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems*, pp. 561–566, 2016.

[12] J. Welde, J. Paulos, and V. Kumar, "Dynamically feasible task space planning for underactuated aerial manipulators," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3232–3239, 2021.

[13] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of the ACM (JACM)*, vol. 40, no. 5, pp. 1048–1066, 1993.

[24] B. Lau, C. Sprunk, and W. Burgard, "Kinodynamic motion planning for mobile robots using splines," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2427–2433, 2009.

[14] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[15] E. D. Markus, "Experimental validation on flatness based control of flexible robot arm," in *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics Intern. Conf.*, pp. 1–5, 2016.

[16] J. Franch, S. K. Agrawal, and V. Sangwan, "Differential flatness of a class of <formula formulatype="inline"> <tex notation="tex">n</tex></formula>-dof planar manipulators driven by 1 or 2 actuators," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 548–554, 2010.

[17] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1st ed., 1994.

[18] S. X. Wei, M. R. Burkhardt, and J. Burdick, "Nonlinear controllability assessment of aerial manipulator systems using lagrangian reduction," in *7th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control*, (Berlin, Germany), pp. 255–260, 2021.

[19] P. Martin, R. Murray, and P. Rouchon, "Flat systems," *Plenary Lectures and Mini-Courses 4th European Control Conf.*, 01 1997.

[20] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *ASME international mechanical engineering congress and exposition*, Citeseer, 1995.

[21] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. ACM*, vol. 40, p. 1048–1066, nov 1993.

[22] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, 2011.

[23] J. Peng, W. Luo, W. Liu, W. Yu, and J. Wang, "A suboptimal and analytical solution to mobile robot trajectory generation amidst moving obstacles," *Autonomous Robots*, vol. 39, 02 2015.

[25] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6235–6240, 2015.

[26] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018.

[27] A. Dixit, M. Ahmadi, and J. W. Burdick, "Risk-averse receding horizon motion planning," *arXiv preprint arXiv:2204.09596*, 2022.

[28] S. X. Wei, A. Dixit, S. Tomar, and J. W. Burdick, "Moving obstacle avoidance: A data-driven risk-aware approach," *IEEE Control Systems Letters*, vol. 7, pp. 289–294, 2023.